IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:

Appellants	:	Bruce W. Melvin et al.
Application No.	:	10/730,390
Filed	:	December 8, 2003
For	:	METHOD AND SYSTEM FOR OUPUT FLOW CONTROL IN NETWORK MULTIPLEXERS

Examiner	:	ELPENORD, Candal
Art Unit	:	2616
Docket No.	:	10991796-2
Date	:	November 10, 2010

APPEAL BRIEF

Mail Stop: Appeal Briefs – Patents
Commissioner of Patents and Trademarks
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This appeal is from the decision of the Examiner, in an Office Action mailed June 1, 2010, rejecting claims 1, 3, 6-8, 10 and objecting to claims 2, 4, 5 and 9.

REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249 Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

RELATED APPEALS AND INTERFERENCES

Appellants' representative has not identified, and does not know of, any other appeals of interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

STATUS OF CLAIMS

Claims 1-10 are pending in the application. Claims 1, 3, 6-8, and 10 were again rejected in an Office Action dated June 1, 2010, in which prosecution was reopened following an initial appeal of final rejections of claims 1, 3, 6-8, and 10. Claims 2, 4, 5 and 9 are objected to as being dependent on a rejected base claim. Appellants appeal the rejection of claims 1, 3, 6-8, and 10, which, along with claims 2, 4, 5, and 9, are copied in the attached CLAIMS APPENDIX.

STATUS OF AMENDMENTS

No Amendment After Final is enclosed with this brief. The last Response was filed July 1, 2009.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent Claim 1

Claim 1 is directed to a method for initiating flow control (lines 3-7 of page 3; lines 6-7 of page 21; lines 13-20 of page 24; line 59 in Figure 14B; and line 106 in Figure 14C) in a network multiplexer (lines 15-26 of page 1; 114 in Figure 1; 700 in Figure 7) that forwards a message descriptor (lines 23-24 of page 1) referencing a communications packet received by a receiving port (702-707 in Figure 7; line 26 of page 8 to line 14 of page 10) to one or more transmit queues (708 in Figure 7; line 26 of page 8 to line 14 of page 10), each transmit queue associated with a transmitting port (702-707 in Figure 7; line 26 of page 8 to line 14 of page 10) which transmits communications packets queued to the transmit queue, the method comprising: (1) providing each transmitting port in the network multiplexer with a high threshold (lines 7-10 of page 15; 1314 in Figure 13A) and a low threshold (lines 7-10 of page 15; 1316 in Figure 13A); and (2) when a message descriptor is queued to a transmit queue associated with a transmitting port, (2a) when the transmit queue currently contains a

maximum number of message descriptors, discarding the message descriptor (lines 27-29 of page 15), and (2b) when the transmit queue currently contains a number of message descriptors equal to or greater than the high threshold of the associated transmitting port, sending a flow control request (line 29 of page 15 to line 3 of page 16) to the receiving port that received the communications packet referenced by the queued message descriptor.

Independent Claim 6

Claim 6 is directed to a network multiplexer (lines 15-26 of page 1; 114 in Figure 1; 700 in Figure 7) system that links physically separate network media by forwarding packets received from each network medium to a number of network media, the network multiplexer system comprising: (1) a number of ports (702-707 in Figure 7; line 26 of page 8 to line 14 of page 10), each port having a transceiver and a communications controller; (2) a memory (712 and 714 in Figure 7; lines 2-5 of page 9; 324 in Figure 3; line 22 of page 5 to line 6 of page 6); (3) an internal bus for transferring packets from ports to memory and from memory to ports (322 in Figure 3; line 22 of page 5 to line 6 of page 6); (4) a receive queue (710 in Figure 7; line 26 of page 8 to line 14 of page 10) and a transmit queue (708 in Figure 7; line 26 of page 8 to line 14 of page 10) associated with each port that contain message descriptors (lines 23-24 of page 1) that each references a communications packet stored in memory (714 in Figure 7; lines 2-5 of page 9); (5) a high threshold (lines 7-10 of page 15; 1314 in Figure 13A) and a low threshold (lines 7-10 of page 15; 1316 in Figure 13A) associated with each transmit queue; (6) an indication of ports to which flow control requests (line 29 of page 15 to line 3 of page 16) have been made associated with each port (1318-1320 in Figure 13A; lines 18-26 of page 15); and (7) an indication of the number of flow control requests made to a port associated with each port (line 29 of page 15 to line 3 of page 16).

Dependent Claim 7

Claim 7 is directed to the network multiplexer (lines 15-26 of page 1; 114 in Figure 1; 700 in Figure 7) of claim 6 wherein, when a message descriptor (lines 23-24 of page 1) is forwarded to a port (702-707 in Figure 7; line 26 of page 8 to line 14 of page 10) for transmission, and when the transmit queue (708 in Figure 7; line 26 of page 8 to line 14 of page 10) of the port is full, the message descriptor is dropped.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. The rejection of claims 1, 3, and 7 under 35 U.S.C. §103(a) as being unpatentable over Bubenik, U.S. Patent No. 5,933,429 ("Bubenik") in view of Wu et al., U.S. Patent No. 5,165,021 ("Wu").

2. The rejection of claims 6, 8, and 10 under 35 U.S.C. §103(a) as being unpatentable over Bubenik.

3. The rejection of claim 7 under 35 U.S.C. §103(a) as being unpatentable over Bubenik in view of Wu.

ARGUMENT

Claims 1-10 are pending in the current application. In an office action dated June 1, 2010 ("Office Action"), the Examiner re-opened prosecution, rejected claims 1, 3, and 7 under 35 U.S.C. §103(a) as being unpatentable over Bubenik et al., U.S. Patent No. 5,933,429 ("Bubenik") in view of Wu et al., U.S. Patent No. 5,165, 021 ("Wu"), and rejected claims 6, 8, and 10 as being unpatentable over Bubenik. In addition, the Examiner indicated conditional allowance of claims 2, 4-5, and 9. Appellants wish to thank the Examiner for the conditional allowance of claims 2, 4-5, and 9, but continue to respectfully traverse the rejections of claims 1, 3, and 6-8.

ISSUE 1

1. The rejection of claims 1 and 3 under 35 U.S.C. §103(a) as being unpatentable over Bubenik in view of Wu.

In the previously filed Appeal Brief, Appellants characterized the currently claimed method and system embodiments of the present invention as follows:

> Claim 1, a method claim, includes the step "providing each transmitting port in the network multiplexer with a high threshold and a low threshold." ... The high and low thresholds associated with transmit queues are discussed beginning on line 7 of page 15 of the current application, and in the paragraph that begins on line 27 of page 15 of the current application. Both the low and high threshold contain threshold values related to the number of message

> descriptors queued to the transmit queue. As discussed in the paragraph that begins on line 27 of page 15:
>
> > In the method of the present invention, when a source attempts to queue a message descriptor to a transmit queue, and the transmit queue is full, then the message descriptor is simply discarded. When a source attempts to queue a message descriptor to a transmit queue already containing a number of message descriptors greater than the high threshold, then the transmit queue sends a flow control directive to the source to direct the source to employ hardware or protocol-level flow control procedures in order to temporarily prevent reception of additional communications packets by the source. When the number of message descriptors queued within the transmit queue has equaled or exceeded the high threshold value, and then falls below one less than the high threshold value, then a source may queue a message descriptor to the transmit queue without receiving a flow control directive. When the number of message descriptors in a transmit queue has equaled or exceeded the high threshold value, and the number of entries has fallen below the low threshold value, then the transmit queue sends release flow control messages to any sources to which the transmit queue had sent flow control messages during the time when the number of queued message descriptors equaled or exceeded the high threshold. However, a transmit queue will not release sources from flow control until the number of queued message descriptors falls below the low threshold.
>
> Thus, both the low threshold and the high threshold describe numbers of queued entries, the low threshold containing a value less than the value contained in the high threshold. The high threshold value is less than the maximum number of entries in the queue, to allow for queuing of some number of message descriptors that arrive after flow control is invoked, when the number of entries exceeds, or is equal to, the high threshold, depending on the implementation, as discussed in the paragraph beginning on line 6 of page 18.

As clearly stated in the above-quoted passage from the previously-filed Appeal Brief, in the current claims, each transmit port and associated transmit queue within a network multiplexer are associated with three different values: (1) a maximum number of message descriptors that can be stored in the transmit queue; (2) a high threshold; and (3) a low threshold. The high threshold is used to determine when flow-control directives are transmitted to a receiving port, to prevent further attempts by the receiving port to queue message descriptors to the transmit queue, and the low threshold is used to determine when a release-flow-control message is sent to a receiving port, so that the receiving port can resume attempts to queue

message descriptors to the transmit queue. When a transmit queue is full, then message descriptors are discarded. When a transmit queue contains a number of queued message descriptors equal to or greater than the high threshold value, any source from which a message is received for queuing to the transmit queue receives a flow-control directive. When the number of message descriptors queued to the transmit queue fall below the low-threshold value, then flow control is released from all sources that were flow controlled as a result of transmitting messages for queuing to the transmit queue.

Independent claim 1 of the current application is reproduced below, for the reader's convenience:

> 1. A method for initiating flow control in a network multiplexer that forwards a message descriptor referencing a communications packet received by a receiving port to one or more transmit queues, each transmit queue associated with a transmitting port which transmits communications packets queued to the transmit queue, the method comprising:
>
> providing each transmitting port in the network multiplexer with a high threshold and a low threshold;
>
> when a message descriptor is queued to a transmit queue associated with a transmitting port,
>
> when the transmit queue currently contains a maximum number of message descriptors, discarding the message descriptor, and
>
> when the transmit queue currently contains a number of message descriptors equal to or greater than the high threshold of the associated transmitting port, sending a flow control request to the receiving port that received the communications packet referenced by the queued message descriptor.

The language of claim 1 can be seen to include a step of "providing each transmitting port in the network multiplexer with a high threshold and a low threshold," as discussed above, and the high threshold clearly indicates a number of queued message descriptors. In the next element of claim 1, "when a message descriptor is queued to a transmit queue associated with the transmitting port" and "when the transmit queue currently contains a maximum number of message descriptors," then the message descriptor is discarded. Thus, a third value associated with a transmit queue, the maximum number of message descriptors that can be contained by the transmit queue, is claimed. Finally, when the transmit queue does not contain a maximum number of message descriptors, but "when the transmit queue currently contains a number of message descriptors equal to or greater than the high threshold of the associated transmitting port," a flow-control request is sent to the receiving port that received the communications message referenced by the queued message descriptor. Claim 1 therefore

corresponds to the above-provided description of the high threshold, namely that the high threshold describes a number of queued entries.

In rejecting claim 1, the Examiner asserts:

> providing each transmitting port in the network multiplexer with a high threshold (see, XOFF feedback message as the high threshold used for halting transmission from the input to the output port buffer, col. 3, lines 49-65) and a low threshold (see, XON threshold as the low threshold, col. 4, lines 10-23); when a message descriptor is queued to a transmit queue associated with a transmitting port (see, data cell identified by pointer, link number and port number indicative of the output queue descriptor, col. 7, lines 8-16, col. 3, lines 34-44), and when the transmit queue currently contains a number of message descriptors equal to or greater than the high threshold of the associated transmitting port, sending a flow control request to the receiving port that received the communications packet referenced by the queued message descriptor (see, XOFF feedback messages as the high threshold used for halting transmission from the input to the output port buffer, col. 3, lines 49-65.

This assertion makes no sense. First of all, the Examiner is attempting to read the claim language "high threshold" onto Bubenik's XOFF feedback message. However, as discussed above, and as clearly claimed in claim 1, the phrase "high threshold" refers to an indication of a number of message descriptors queued to a queue, not to a feedback message. For this reason alone, the Examiner's rejection of claim 1 must certainly fail, because the Examiner is attempting to read "high threshold," an indication of a number of message descriptors queued to a transmit queue, onto a feedback message that is sent, in Bubenik, by a communication path to an input port processor.

The portion of Bubenik cited by the Examiner, and additional portions surrounding the cited portion, including the lines of Bubenik beginning with line 9 of column 3 to line 23 of column 4:

> In order to traverse the switch 1, a data cell 22 first enters the switch 1 on a link 24 to an input port processor 14 and is buffered in a queue 26 of input buffers. The data cell 22 is then transmitted from the queue 26 of input buffers through the Data Crossbar 10 to a queue 28 of output buffers in an output port processor 16. From the queue 28 of output buffers, the data cell 22 is transmitted onto a link 30 outside of the switch 1 to, for example, another switch.
>
> To facilitate traversal of the switch 1, each input port processor 14 includes a cell buffer RAM 32 and each output port processor 16 includes a cell buffer RAM 34. The cell buffer RAM's 32 and 34 are organized into the respective input and output queues 26 and 28. All data cells 22 in a connection pass must through a unique input queue 26 and a unique output

queue 28 for the life of the connection. The queues 26 and 28 thus preserve cell ordering. This strategy also allows quality of service ("QoS") guarantees on a per connection basis.

Three communication paths are used to facilitate traversal of the switch 1 via probe and feedback messages: a Probe Crossbar 42, an XOFF Crossbar 44, and an XON Crossbar 46. The Probe Crossbar 42, which in this particular embodiment is an NxN crosspoint switch, is used to transmit a multiqueue number from an MTC 18 to an output port processor 16. Each input port processor 14 includes a plurality of scheduling lists 47, each of which is a circular list containing input queue numbers for a particular connection. Each multiqueue number is derived from information provided to the MTC 18 from a scheduling list 47 in an input port processor 14. A multiqueue number identifies one or more output queues 28 to which a data cell may be transmitted when making a connection. An output port processor 16 uses the multiqueue number to direct a request message probe to the appropriate output queue or queues 28 and thereby determine if there are enough output buffers available in the output queue or queues 28 for the data cell.

The XOFF Crossbar 44, which in this particular embodiment is an NxN crosspoint switch, is used to communicate "DO NOT SENT" type feedback messages from an output port processor 16 to an input port processor 14. The XOFF feedback messages are asserted to halt the transmission of request message probes through the Probe Crossbar 42 from an input port processor 14 to an output port processor 16, and thus put a scheduling list 47 within the receiving input port processor 14 in an XOFF state, meaning that the scheduling list 47 cannot be used to provide a multiqueue number. The scheduling list 47 remains in an XOFF state until receiving an XON message from the output port processor 16, as described below. An input port processor 14 responds to an asserted XOFF feedback message by modifying XOFF state bits in a descriptor of the scheduling list 47. The XOFF state bits prevent the input port processor 14 from attempting to send a request message probe from the input port processor 14 to the output port processor 16 until notified by the output port processor 16 that output buffers are available for a corresponding connection.

The "DO NOT SEND" type feedback messages also halt the transmission of data cells from an input port processor 14 to an output port processor 16 when sufficient buffer space is not available to receive data cells in the output port processor 16. In such a case, an input port processor 14 will not transmit any data cells through the Data Crossbar 10. An idle cell, containing a complemented cyclic redundancy check (CRC) calculation, is transmitted instead.

The XON Crossbar 46, which in this particular embodiment is an NxN crosspoint switch, is used to communicate "ENABLE SEND" type feedback messages from an output port processor 16 to an input port processor 16. More particularly, the XON Crossbar 46 communicates an XON feedback

> message from an output port processor 16 to an input port processor 14. When an XOFF feedback message has been asserted by an output port processor 16 in response to a request probe message from an input port processor 14, the output port processor 16 sets a state bit in a queue descriptor of a corresponding output queue 28. When the number of data cells in that output queue 28 drops below an XON threshold, an XON message is sent from that output port processor 16 to the input port processor 14. The XON message enables the scheduling list 47 in the input port processor 14 to be used in the sending of request probe messages, and hence data cells.

As can be seen in figure 1 of Bubenik, and as described in the above-quoted passage, Bubenik's network switch includes four distinct and separate communications paths. A data path (10 in Figure 1 of Bubenik) is used to transmit data cells, or messages, from input buffers to output buffers. A probe communications path (42 in Bubenik) transmits request message probes sent by an input port processor to output port processors in order to determine whether there is sufficient output buffer capacity to allow a next data cell or data cells to be forwarded to the appropriate output buffers. In response to receiving these request message probes, an output port processor can, when there is insufficient buffering capacity in the output transmit buffers, send a "DO NOT SEND" feedback message through the XOFF communications path (44 in Figure 1 of Bubenik) to the input processors so that the input processors do not send any more request message probes or data cells to the target output transmit buffer or buffers. Thus, the XOFF feedback message is a flow-control request that is sent when there is insufficient space on the transmit side for buffering a data cell. An output port processor can determine when enough transmit buffers are subsequently freed up, after XOFF flow control has been invoked, to allow resumption of data-cell forwarding to transmit ports by comparing the number of data cells in an output queue to the XON threshold. As explicitly stated by Bubenik, beginning on line 17 of column 4:

> When the number of data cells in that output queue 28 drops below an XON threshold, an XON message is sent from that output port processor 16 to the input port processor 14. The XON message enables the scheduling list 47 in the input port processor 14 to be used in the sending of request probe messages, and hence data cells.

Thus, when the number of message queues drops below the XON threshold, an XON message is sent to reverse the effects of a previously sent previous XOFF message.

Bubenik clearly describes two values associated with output transmit queues. The first value is a space-available value that serves a similar purpose as the currently

claimed "maximum number of message descriptors" value. The second value, the XON threshold, is related to the currently claimed "low threshold" value that is discussed, in greater detail, in claim 2. Bubenik does not teach, mention, or even remotely suggest the currently claimed "high threshold," which is less than the maximum number of message descriptors that can be queued to a transmit queue. The Examiner attempts to read "high threshold" onto Bubenik's XOFF feedback messages, but, as discussed above, Bubenik's XOFF feedback messages are messages sent through the XOFF communications path, and are not equivalent to, or even remotely related to, a threshold number of message descriptors of any kind, let alone the currently claimed high threshold. Bubenik sends an XOFF feedback message when there is insufficient space in the transmit queues to accommodate another data cell but, by contrast, according to the currently-claimed method, a flow control request is sent when a transmit queue contains a number of message descriptors equal to or greater than the high threshold. Clearly, the high threshold is less than the maximum number of message descriptors that can be queued to a message queue, because, otherwise, the number of message descriptors queued to the transmit queue could not exceed the high threshold, but claim 1 states that the number of message descriptors queued to a transmit queue can equal or exceed the high threshold. Thus, Bubenik sends a flow-control message when there is insufficient space in the transmit queues to receive a next data cell, while the currently-claimed message sends a flow-control request when there is more space available on the transmit queue for message descriptors, but the number of message descriptors on the transmit queue is equal to or greater than the high threshold. Furthermore, the currently-claimed method discards a message when the number of message descriptors queued to a transmit queue is equal to the maximum number of message descriptors that can be queued to the transmit queue, while Bubenik does not teach, mention, or suggest discarding messages or message descriptors.

To summarize, Bubenik describes a network switch in which only two values are associated with output ports, a value indicating the maximum space available within the output queues and an XON threshold which indicates a number of data cells queued within a transmit queue below which a flow-control message is sent to input ports to release flow control established by previously sending an XOFF feedback message to the input port. By contrast, as explained above, and as explained in previously filed Responses and a previously filed Appeal Brief, the currently claimed method associates three different values with a transmitting port, including a high threshold, a low threshold, and a maximum number of

message descriptors that can be queued to the transmit queue associated with that port. In the currently claimed method, when a message is received by the transmit port and the queue associated with the transmit port contains the maximum number of message descriptors, the received message is discarded. Nothing in Bubenik teaches, mentions, or suggests this discarding step. In the currently claimed method, when a message is received for queuing for the is equal to or greater than the high threshold, the message is queued, but a flow-control request is returned to the input port so that the input port does not forward any further messages to the output port. There is no analogy or equivalent to the high threshold taught, mentioned, or even remotely suggested in Bubenik for the currently claimed high threshold. In the currently claimed method, when the number of message descriptors in a transmit queue falls below the low threshold of the associated transmitting port, "but the number of message descriptors contains in the transmit queue exceeded or equaled the high threshold used of the associated transmitting port more recently than the number of message descriptors contains in the transmit queue was equal to the low threshold of the associated transmitting port," a flow-control request is sent to the receiving port in order to release flow control and allow the receiving port to continue sending messages to the transmit port. Bubenik's XON threshold is analogous, but the comparison step in Bubenik only considers the XON threshold, and not both a high threshold, a low threshold, and the history of the number of message descriptors queued to the transmit queue, as does the currently-claimed method.

The Examiner's attempt to read the currently-claimed high threshold onto Bubenik's triggering of the sending of XOFF feedback messages must fail, because Bubenik sends XOFF messages only when the transmit queues are too full to accommodate another data cell, while, in the currently-claimed method, the high threshold is a number less than the maximum number of messages descriptors that can be queued to a transmit queue by at least 2 message descriptors. It is the intent of the currently claimed method to allow some number of additional message descriptors to be queued to a transmit queue after the high threshold is reached, as discussed in the current application in the paragraph that begins on line 6 of page 18. There is nothing equivalent to this soft flow-control point taught, mentioned, or suggested in Bubenik. By contrast, were the Examiner to attempt to read the currently claimed maximum number of messages associated with a transmit queue onto triggering of an XOFF feedback message in Bubenik, that attempt would also fail, since the currently-claimed method discards a message when the number of message descriptors queued to a transmit queue equals the maximum number of message descriptors that can be queued to the

transmit queue, and Bubenik does not, and does not need to, discard messages. There is simply no way to read the three values associated with a transmit queue by the currently claimed method onto the two values used by Bubenik to toggle feedback control via XOFF and XON messages.

On page 5 of the Office Action, the Examiner states:

> However, Wu '021 from a similar field of endeavor discloses the above claimed features: Regarding claim 1, when the transmit queue currently contains a maximum number of message descriptors (see, the number of the number of queue locations being greater than loadsheding value associated with packet descriptor, col. 13, lines 46-60), discarding the message descriptor (see, discarding the packet descriptor, col. 13, lines 61-64) when the number of free blocks greater than a loadsheding factor by comparing the number of free blocks with the loadsheding factor, col. 2, lines 30-38, lines 48-58, the loadsheding represent an amount of free space in the transmit queue, col. 2, lines 66-68, the transmit queue as link list, col. 3, lines 5-6.

Appellants' representative believes that this statement rather seriously mischaracterizes Wu's loadsheding value. As explicitly stated by Wu beginning on line 14 of column 2, and as again stated by Wu beginning on line 63 of column 2:

> A loadsheding value correlating to the packet's destination port and priority is determined and compared with a measure of the amount of free space in the transmit queue.

The loadsheding value of Wu is not, and has nothing to do with the number of message descriptors queued to a transmit queue, but is instead a value determined from a packet's destination port and a priority. Furthermore, Wu does not state, intimate, or even remotely suggest that packets are dropped when a transmit queue contains a maximum number of message descriptors, but instead explicitly states, beginning on line 18 of column 2, that a packet is discarded when the loadsheding value, which is computed based on a priority and destination port, is greater than the measure of the amount of free space in the transmit queue. By contrast, claim 1 refers to a message descriptor being discarded "when the transmit queue currently contains a maximum number of message descriptors."

Moreover, the Examiner makes a very serious error in attempting to combine two very different protocols described by Bubenik and Wu. As those familiar with networking and communications protocols well understand, any change to a protocol is fraught with potential problems and errors. Protocols are carefully evaluated to detect various types of pathological behaviors when even the smallest changes to a protocol are

attempted. The Examiner suggests that one could simply graft discarding of messages, as taught by Wu, onto Bubenik's protocol in order to arrive at the currently claimed method. As discussed above, Wu does not teach, mention, or suggest that for which it is cited, and no combination of Wu and Bubenik produces the three values associated in the current method with each transmit queue. Thus, the combination fails to teach or suggest the currently claimed invention. However, the attempt to casually combine two dissimilar protocols should, in and of itself, result in failure of the obviousness-type rejection, because the Examiner has not undertaken even cursory evaluation of the viability of discarding messages in Bubenik's protocol, which does not contemplate discarding of messages. In fact, Bubenik's network switch uses three out-of-band communications paths in order to avoid the need for discarding messages. In Bubenik's network switch, a receiving port employs the probe communications path in order to determine, in advance of sending a data cell, whether or not transmit queues can receive the data cell. There is no reason for discarding messages in Bubenik, and there is no reason for proposing that messages be discarded. The Examiner has failed to point to dropped-message-recovery facilities in Bubenik's protocol that would allow messages to be dropped. There is no reason for combining Bubenik and Wu and there is no reason to believe that the combination of Bubenik and Wu would produce a correctly functioning protocol. Because protocols are known to be highly fragile and difficult to modify, there is every reason to believe that any attempt to discard messages in Bubenik would lead to serious problems.

The Examiner's stated justification for the combination of Bubenik and Wu is: "The motivation would have been to provide traffic policing by dropping the packet or frame descriptor when the transmit queue threshold is exceeded." This statement makes no sense. Bubenik does not employ a high threshold for the number of data cells that can be queued to a transmit queue, so it is unclear to what threshold the Examiner is referring. Bubenik uses out-of-band communications, via the probe and XOFF communications paths, to ensure that no data cell is sent to transmit ports when the transmit buffers associated with transmit ports cannot accommodate the data cell. Thus, there is no need in Bubenik to discard data cells. The term "traffic policing" has no meaning in network communications. Finally, as discussed above, one cannot simply discard messages unless there are robust dropped-message recover facilities. The Examiner fails to even consider whether or not Bubenik's protocol could tolerate data-cell discards, even though, as pointed out above, Bubenik ensures that data cells are not transmitted to transmit queues when the transmit queues have

insufficient space to accommodate them.

ISSUE 2

2. <u>The rejection of claims 6, 8, and 10 under 35 U.S.C. §103(a) as being unpatentable over Bubenik.</u>

Claim 6, and claims 8 and 10, which depend from claim 6, are directed to a network multiplexer system in which each transmit queue is associated with a high threshold and a low threshold, in which the network multiplier system includes "an indication of ports to which flow-control requests have been associated with each port" and "an indication of a number of flow-control requests made to a port associated with each port." The rejection of claim 6 closely parallels the rejection of claim 1, and fails for the same reason as the rejection of claim 1. As an example, the Examiner again attempts to read the claim language "high threshold" onto an XOFF feedback message. As discussed above with respect to the preceding issue, the "high threshold" is an indication of a number of message descriptors queued to a transmit queue, and has nothing to do with feedback message. The final two elements of claim 6 are also rejected based on XOFF feedback messages. However, the second-to-last element of claim 6 in an indication of ports to which flow-control requests have been made and the final element of claim 6 is an indication of the number of flow-control requests made to a port associated with each port." There is nothing in column 3 or 4 of Bubenik which teaches, mentions, or even remotely suggests anything that keeps track of the number of flow-control requests made to a port. In Bubenik, there is no reason to keep track of such a number, since Bubenik maintains the receiving ports in one of an XON or XOFF state. There is no consideration, anywhere discussed in Bubenik, for changing the XON state to an XOFF state or vice versa by considering the number of flow-control requests that have been made to a port. The currently claimed "high threshold" is not a feedback message, and simply referring to Bubenik's XOFF feedback messages for teaching the final three elements of claim 6, which are neither taught, mentioned, or even remotely suggested in Bubenik, does not constitute finding a teaching or disclosure of these three elements. Clearly, claim 6 is not taught or suggested by Bubenik's disclosure, and therefore no claim that depends from claim 6, including claims 7-10, can possibly be taught or suggested by Bubenik.

ISSUE 3

2. The rejection of claim 7 under 35 U.S.C. §103(a) as being unpatentable over Bubenik in view of Wu.

In the rejection of claim 7, the Examiner relies on Bubenik disclosing all of the claim limitations except "when a message descriptor is forwarded to a port for transmission, and when the transmit queue of the port is full, the message descriptor is dropped." As discussed above with respect to Issue 2, Bubenik does not disclose all of the claim limitations of claim 6, including the final three elements of claim 6, from which claim 7 depends. As discussed above with respect to Issue 1, Wu does not discard messages when a transmit queue is full, but instead computes a loadsheding factor, based on a priority and destination port, and compares the computed loadsheding factor to amount of free space in the transmit queue. As further discussed above with respect to Issue 1, there is no reason, motivations, or justification for combining Wu discarding of packets with Bubenik's protocol, in which out-of-band communications is employed to prevent data cells from being sent to transmit queues that cannot accommodate them, and which therefore does not provide for discarding of data cells.

CONCLUSION

The currently claimed method and network multiplexer system associates three different values with each transmit queue, including a maximum number of queue entries, a high threshold, and a low threshold. Bubenik associates two values with transmit queues, rather than three, and uses the two values in a different way than the three values are claimed to be used in the current claims. The currently claimed network multiplexer system includes indications of ports to which flow-control requests have been made and an indication of the number of flow-control requests made to a port, while in Bubenik's binary OFF/ON system, maintenance of such information is not taught, suggested, or in any way needed. Bubenik clearly does not teach, mention, or suggest either the claimed method or the claimed network multiplexer system of the current application. Wu does not teach, mention, or suggest that for which it is cited. Wu discloses discarding packets based on comparing port identities and priority levels, encapsulated in loadsheding values, to available free space.

By contrast, the currently claimed method and network multiplexer system discards messages when a maximum number of message descriptors have been queued to a transmit queue. The protocols of Bubenik and Wu, despite the Examiner's assertion to the contrary, cannot be casually combined, as discussed with regard to the first issue, above. Any proposed change to a protocol involves careful analysis, including determining whether or not a proposed new behavior or feature can be accommodated by the system implementing the protocol. Bubenik's is designed to avoid discarding of data cells, and does not provide any indication that discarded data cells can be recovered. Furthermore, Bubenik employs out-of-band communications paths to avoid ever sending a data cell to a transmit queue that cannot accommodate queuing the data cell, so there is no need for discarding messages in Bubenik. The Examiner's proposed combination of Bubenik and Wu is unjustified, unsupported, and almost certainly unsupportable.

Appellants respectfully submit that all statutory requirements are met and that the present application is allowable over all the references of record. Therefore, Appellants respectfully request that the present application be passed to issue.

Respectfully submitted,
Bruce W. Melvin et al.
OLYMPIC PATENT WORKS PLLC

By ______________________
Robert W. Bergstrom
Registration No. 39,906

CLAIMS APPENDIX

1. A method for initiating flow control in a network multiplexer that forwards a message descriptor referencing a communications packet received by a receiving port to one or more transmit queues, each transmit queue associated with a transmitting port which transmits communications packets queued to the transmit queue, the method comprising:

providing each transmitting port in the network multiplexer with a high threshold and a low threshold;

when a message descriptor is queued to a transmit queue associated with a transmitting port,

when the transmit queue currently contains a maximum number of message descriptors, discarding the message descriptor, and

when the transmit queue currently contains a number of message descriptors equal to or greater than the high threshold of the associated transmitting port, sending a flow control request to the receiving port that received the communications packet referenced by the queued message descriptor.

2. The method of claim 1 further including:

when a message descriptor is queued to a transmit queue associated with a transmitting port,

when the transmit queue currently contains a number of message descriptors greater than or equal to the low threshold of the associated transmitting port, but the number of message descriptors contained in the transmit queue exceeded or equaled the high threshold of the associated transmitting port more recently than the number of message descriptors contained in the transmit queue was equal to the low threshold of the associated transmitting port, sending a flow control request to the receiving port that received the communications packet referenced by the queued message descriptor.

3. The method of claim 1 further including:

when a transmitting port transmits a packet referenced by a message descriptor to a destination port,

releasing the message descriptor, and

when the destination port currently contains a number of queued message descriptors equal to one less than the destination port's low threshold, sending a release flow control request to any receiving ports to which a flow control request was sent while the transmit queue contained a number of message descriptors equal to or greater than the high threshold of the associated transmitting port.

4. The method of claim 2 further including:

when a transmitting port transmits a packet referenced by a message descriptor to a destination port,

releasing the message descriptor, and

when the destination port currently contains a number of queued message descriptors one less than the destination port's low threshold, sending a release flow control request to any receiving ports to which a flow control request was sent while the transmit queue contained a number of message descriptors greater than or equal to the low threshold of the associated transmitting port.

5. The method of claim 4 further including:

when a receiving port is flow controlled and receives a number of release flow control requests equal to the number of received flow control requests,

releasing flow control by the receiving port.

6. A network multiplexer system that links physically separate network media by forwarding packets received from each network medium to a number of network media, the network multiplexer system comprising:

a number of ports, each port having a transceiver and a communications controller;

a memory;

an internal bus for transferring packets from ports to memory and from memory to ports;

a receive queue and a transmit queue associated with each port that contain message descriptors that each references a communications packet stored in memory;

a high threshold and a low threshold associated with each transmit queue;

an indication of ports to which flow control requests have been made associated with each port; and

an indication of the number of flow control requests made to a port associated with each port.

7. The network multiplexer of claim 6 wherein, when a message descriptor is forwarded to a port for transmission, and when the transmit queue of the port is full, the message descriptor is dropped.

8. The network multiplexer of claim 6 wherein, when a message descriptor is forwarded to a port for transmission, and when the transmit queue of the port contains a number of message descriptors greater than or equal to the high threshold associated with the port, a flow control request is sent to the port that received the communications packet referenced by the message descriptor and a indication that a flow control request has been sent to the port that received the communications packet is saved by the port to which the message descriptor is forwarded.

9. The network multiplexer of claim 6 wherein, when a message descriptor is forwarded to a port for transmission, and when the transmit queue of the port has contained a number of message descriptors greater than or equal to the high threshold associated with the port more recently than the transmit queue of the port has contained a number of message descriptors less than the low threshold associated with the port, a flow control request is sent to the port that received the communications packet referenced by the message descriptor and a indication that a flow control request has been sent to the port that received the communications packet is saved by the port to which the message descriptor is forwarded.

10. The network multiplexer of claim 6 wherein, when a port removes a message descriptor from the transmit queue associated with the port, and when the number of messages contained in the transmit queue currently equal one less than the low threshold associated with the port, a release flow control message is sent to each port referenced by indications saved by the port.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.